UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/815,052	03/31/2004	Jesper Kiehn	M61.12-0615	7946

27366 7590 12/23/2010

WESTMAN CHAMPLIN (MICROSOFT CORPORATION)
SUITE 1400
900 SECOND AVENUE SOUTH
MINNEAPOLIS, MN 55402

EXAMINER	
HOFFMAN, BRANDON S	

ART UNIT	PAPER NUMBER
2433	

MAIL DATE	DELIVERY MODE
12/23/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte JESPER KIEHN, PAVEL HRUBY, and GEIR OLSEN

Appeal 2009-007931
Application 10/815,052[1]
Technology Center 2400

Before KENNETH W. HAIRSTON, MARC S. HOFF,
and ELENI MANTIS MERCADER, *Administrative Patent Judges*.

HOFF, *Administrative Patent Judge*.

DECISION ON APPEAL[2]

---

[1] The real party in interest is Microsoft Corporation.
[2] The two-month time period for filing an appeal or commencing a civil action, as recited in 37 C.F.R. § 1.304, or filing a request for rehearing, as recited in 37 C.F.R. § 41.52, begins to run from the "MAIL DATE" (paper delivery mode) or the "NOTIFICATION DATE" (electronic delivery mode) shown on the PTOL-90A cover letter attached to this decision.

STATEMENT OF THE CASE

Appellants appeal under 35 U.S.C. § 134(a) from a Final Rejection of claims 1, 3-18, 20-34, and 36-39. We have jurisdiction under 35 U.S.C. § 6(b).

We affirm.

Appellants' invention relates to a method of providing Resource-Event-Agent (REA) model based security (Abstract). The REA model is a prescriptive accounting model that may be implemented in an object-oriented software programming language that enables model driven software development for business users using familiar accounting concepts (Spec. 1:10-22, 2:13-15, and 7:6-15). As a first step in the method, an association is identified between a first object and a second object, where the first object is an Agent type and the second object is any REA object (Abstract). A security policy association is assigned to the identified association that defines the properties of security between the first and second objects (Abstract; Spec. 23:21-24:6). Finally, the security policy association object is stored on a computer storage medium (Abstract).

Claim 1 is exemplary:

1. A method of providing Resource-Event-Agent (REA) model based security, the method comprising:

identifying an REA defined association of a type which dictates ownership between a first object and a second object in an REA model;

creating an association class object for the REA defined association between the first object and the second object, the association class object having properties defining security between the first object and the second object; and

storing the association class object on a computer storage medium for use in providing security between the first object and the second object.

The prior art relied upon by the Examiner in rejecting the claims on appeal is:

Tingey	US 2004/0133583 A1	Jul. 8, 2004
Boozer	US 2004/0205355 A1	Oct. 14, 2004

Claims 1, 3-18, 20-34, and 36-39 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Boozer in view of Tingey.

ISSUES

Appellants contend that none of the references discloses a method of providing Resource-Event-Agent (REA) model based security that includes "creating an association class object for the REA defined association between the first object and the second object, the association class object having properties defining security between the first object and the second object" (App. Br. 9). Appellants assert that Tingey only briefly mentions a general concept of security (App. Br. 10). Appellants further argue that none of the references discloses a control type association or a custody type association (App. Br. 11-12). Appellants contend that Boozer does not state that the association between the first and second objects is imposed by the containment boundary or that "the containment boundary establishes security rules for the two objects, which would be an association class object" (App. Br. 13). Appellants assert that the Examiner's use of the *ipsissimis verbis* principle should not apply (App. Br. 14).

Appellants' contentions present us with the following two issues:

1. Do the references disclose a method of providing Resource-Event-Agent (REA) model based security that includes "creating an association

class object for the REA defined association between the first object and the second object, the association class object having properties defining security between the first object and the second object?"

2. Do the references disclose defining control type associations or custody type associations between a first and a second object?

FINDINGS OF FACT

The following Findings of Fact (FF) are shown by a preponderance of the evidence.

The Invention

1. A "control association" type and "custody association" type are "common" in the conventional REA data model. When an Agent is at one end of an Association of type "Control," the Agent will be granted rights on the Class of Objects at the other end. When an Association between a Resource and an Agent is of type Custody, the Agent will be granted some default permissions on the Resource (Fig.4; Spec. 18:1-5; 20:1-27).

Boozer

2. Boozer discloses a data access security system 60 for accessing information stored within a data storage unit. The stored information is retrieved through resource objects (70, 74, 76, and 78) which are interconnected through a complex set of relationships or associations 72. By definition, associations 72 define conceptual relationships between an instance of one class and an instance of another class (Fig. 2; ¶ [0016]).

3. Boozer discloses that a requester 62 may issue a request for data from resource object 70 by sending the request to a security request handler 64 for examination of whether access should be granted to all or any

of the requested information, as well as what kinds of operations the requester 62 may perform on that information (Fig. 2; ¶ [0017]).

4. Boozer discloses that the security request handler 64 constructs a containment boundary 68 around the resource object 70. The containment boundary 68 defines which objects are containers or parents having access to object 70. Object security rules 66 help to define the containment boundary 68 by specifying what object associations 72 (corresponding to objects 74, 76, and 78) are to be included in constructing the containment boundary 68. (Fig. 2; ¶ [0018]).

5. Boozer discloses that the containment boundary 68 may be established based upon object class where the security rules define how to traverse from an object of a class to the related object with which it is associated. (¶¶ [0019] and [0021]).

6. Boozer discloses an example for implementation of a check authorization process 202 for the security request handler 64 which determines whether to grant a requester permission to access an object. Process 202 includes the creation of "access control objects" that specify whether the system should grant or deny one or more identities access to a specific requested object. These access control objects can further specify what kinds of activities an identity can perform on the requested object. (Figs. 2 and 9; ¶ [0029]).

7. Boozer discloses that specifically during the authorization process 202 that the security request handler 64 verifies if any access controls can be located that are directly attached to requested object. If the security request handler 64 can locate these access controls, it determines whether any of these pertain to the current requester and to the permission

sought by the current requester. If these access controls are unrelated to the current requester, the security request handler 64 verifies if there is a group which the requester is associated with or whether there is specific authorization for that requester. If no group or authorization exists for the current requester, then the object access security rules, stored in "cache" memory, for the requested object are examined (Figs. 2 and 9; ¶ [0029]).

Tingey

8. Tingey disclose an REA model that uses security "based on some form of classification" (¶ [0066]).

PRINCIPLE OF LAW

On the issue of obviousness, the Supreme Court has stated that "the obviousness analysis cannot be confined by a formalistic conception of the words teaching, suggestion, and motivation." *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 419 (2007). Further, the Court stated "[t]he combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results." *Id.* at 416.

ANALYSIS

Claims 1, 3-18, 20-34, and 36-39

We select claim 1 as representative of this group of claims, pursuant to our authority under 37 C.F.R. § 41.37(c)(1)(vii).

Representative claim 1 recites "creating an association class object for the REA defined association between the first object and the second object, the association class object having properties defining security between the

first object and the second object." Independent claims 18 and 34 disclose a claim limitation similar in scope.

We do not consider Appellants' arguments to be persuasive to show Examiner error. We agree with the Examiner's finding that Boozer discloses associations that are assigned to objects based upon class which contain security definitions regarding access (Ans. 3). Boozer discloses a data access security system 60 for accessing information stored within a data storage unit (FF 2). The stored information is retrieved through resource objects (70, 74, 76, and 78) which are interconnected through a complex set of relationships or associations 72 (FF 2). By definition, associations 72 define conceptual relationships between an instance of one class and an instance of another class (FF 2). A requester 62 may issue a request for data from resource object 70 by sending the request to the security request handler 64 for examination of whether access should be granted to all or any of the requested information, as well as what kinds of operations the requester 62 may perform on that information (FF 3). The security request handler 64 constructs a containment boundary 68 around the resource object 70, wherein the containment boundary 68 defines which objects are containers or parents having access to object 70 (FF 4). Object security rules 66 help to define the containment boundary 68 by specifying what object associations 72 are to be included in constructing the containment boundary 68 (FF 4). Containment boundaries may be established for objects by setting up object access security rules *based on object class* that describe how to traverse from an object of a class to the related object with which it is associated (FF 5).

More particularly, Boozer discloses a specific implementation of a check authorization process for the security request handler 64 that includes the creation of *access control objects* which specify whether the system should grant or deny one or more identities access to a specific requested object (FF 6 and 7). These access control objects can further specify what kinds of activities an identity can perform on the requested object (FF 6). Since the containment boundaries may be constructed by setting up a series of rules 66, *based upon class*, it logically follows that the creation of the access control objects may be established based upon class as well (FF 5 and 6). Accordingly, we find that the creation of access control objects based upon class discloses the claimed "association class objects" that define the security properties between a first and second object.

Although Tingey discloses an REA model that uses security *"based on some form of classification,"* (FF 8) the Examiner merely relies upon Tingey to disclose the specific REA data model (Ans. 8, FF 8). Thus, the combined references disclose a method of providing REA model based security that assigns access control objects, based upon class, that include properties defining security between at least two objects. As noted *supra*, we find that the "access control objects" perform a similar, if not the same, function as "association class objects."

Regarding the claim limitations of control type and custody type associations, the Specification discloses that these types of associations are "common" in the conventional REA data model and, thus, are Admitted Prior Art (FF 1).

Therefore, we find that the Examiner has established the prima facie obviousness of the claims, because the combination of Boozer and Tingey

discloses a method of providing Resource-Event-Agent (REA) model based security that includes "creating an association class object for the REA defined association between the first object and the second object, the association class object having properties defining security between the first object and the second object." As a result, we will sustain the Examiner's § 103 rejection of representative claim 1 and that of claims 3-18, 20-34, and 36-39.

CONCLUSIONS

The references disclose a method of providing Resource-Event-Agent (REA) model based security that includes "creating an association class object for the REA defined association between the first object and the second object, the association class object having properties defining security between the first object and the second object."

Control type associations or custody type associations between a first and a second object are admitted prior art features.

ORDER

The Examiner's rejection of claims 1, 3-18, 20-34, and 36-39 is affirmed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

ELD

WESTMAN CHAMPLIN (MICROSOFT CORPORATION)
SUITE 1400
900 SECOND AVENUE SOUTH
MINNEAPOLIS, MN 55402